

Makalah Seminar Tugas Akhir

Fajar Adi Daniarto

Email : Fajar@student.undip.ac.id

ANALISIS PERANGKAT LUNAK IP TELEPHONY PADA JARINGAN DENGAN SISTEM OPERASI LINUX**Abstraksi**

Internet telah melahirkan suatu teknologi baru yang disebut IP Telephony, IP Telephony menggunakan internet untuk mengirimkan suara antar telepon dengan telepon, PC dengan PC, ataupun telepon dengan PC, pada saat yang bersamaan.

H.323 merupakan protokol yang mencakup rekomendasi seri H ITU-T (International Telecommunications Union), yang menjadi sinyal protokol yang telah di standarisasi untuk IP Telephony.

Tujuan Tugas Akhir ini adalah melihat perbandingan kecepatan VoIP pada aplikasi voice chat dengan bahasa Perl. Manfaat dari perbandingan ini untuk mengetahui buffer yang dibutuhkan pada aplikasi tersebut antar dua komputer dalam satu jaringan peer to peer.

Dengan menggunakan dua buah komputer yang bersistem operasi Linux dapat saling berkomunikasi, baik data maupun suara.

Aplikasi voice chat yang digunakan disini, belum menggunakan kompresi, karena itu dicari buffer yang tepat untuk memperkecil delay yang terjadi.

1.1. Latar Belakang

VoIP (*Voice over Internet Protocol*) merupakan teknologi yang menawarkan solusi komunikasi suara melalui jaringan paket (*IP Network*). Teknologi ini menjanjikan suatu kelebihan, sehingga banyak pihak yang ikut mengembangkan.

Pada prinsipnya teknologi VoIP menggunakan *Internet Protocol* sebagai infrastruktur transfer data (yang berupa suara). Selama ini pada umumnya komunikasi suara untuk bercakap-cakap menggunakan *Public Switched Telephone Network* (PSTN).

Ada dua hal pokok yang membuat VoIP merupakan alternatif cantik dibandingkan PSTN. Pertama, secara bisnis, komunikasi jarak jauh yang dilakukan melalui PSTN harus melalui SLJJ (Sambungan Langsung Jarak Jauh) atau SLI (Sambungan Langsung Internasional), yang membebankan ongkos yang tinggi.

Kedua, secara teknologi, VoIP relatif lebih hemat *bandwidth* karena kemampuan kompresinya. Transfer data melalui IP menggunakan teknologi jaringan *packet-switched*, sementara PSTN merupakan jaringan *circuit-switched*. Pada *packet-switched*, data dipecah menjadi paket-paket kecil, kemudian dikirim melalui kanal yang berbeda antara pengirim dengan tujuan akhirnya. Untuk itu setiap paket harus mempunyai *header* untuk identifikasi dan pengurutan data di tujuan.

1.2 Tujuan

Tujuan Tugas Akhir ini adalah menganalisis perangkat lunak IP Telephony pada perbandingan kecepatan VoIP dengan *buffer*. Manfaat dari perbandingan ini untuk mengetahui *buffer* yang dibutuhkan pada aplikasi *voice chat*

antar dua komputer dalam satu jaringan secara langsung.

1.3 Pembatasan Masalah

Permasalahan yang dibahas pada Tugas Akhir ini adalah menganalisa performansi IP Telephony dalam suatu jaringan Linux. Dalam pembuatan aplikasi *voice chat* ini, bahasa pemrograman yang dipakai adalah bahasa Perl. Bahasa Perl adalah sebuah bahasa yang berorientasi jaringan.

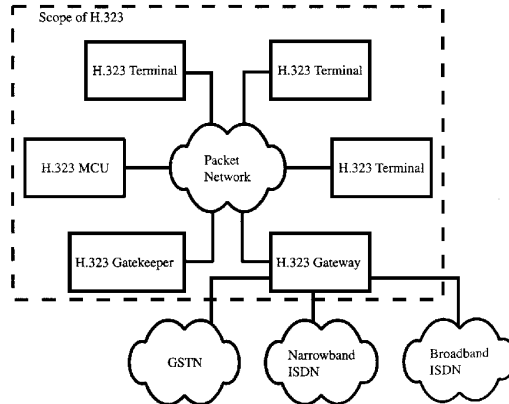
- Untuk membatasi perangkat lunak ini, maka diambil batasan sebagai berikut :
 - Bahasa pemrograman yang digunakan adalah PERL (*Practical Extraction and Reporting Language*).
- Pengujian dilakukan pada jaringan *peer to peer*.
- Pengujian yang dilakukan adalah mencatat *delay* pada *buffer* yang besarnya berbanding 1:1, 1:2, 2:2, 2:4, 4:4, 4:8, 8:8, 8:16, 16:16, dalam satuan KB.
- Pengukuran *delay* dilakukan secara manual dengan menggunakan *stopwatch*, karena program ini tidak menyediakan fasilitas secara otomatis.
- Dalam program ini tidak menggunakan perhitungan secara otomatis, karena perintah yang digunakan tidak menyediakan dukungan untuk itu.
- Dalam program ini tidak memperhitungkan QoS (*Quality of Service*).
- Pengujian dilakukan pada sistem operasi Linux Suse 9.0.

2. DASAR TEORI

2.1 Arsitektur H.323

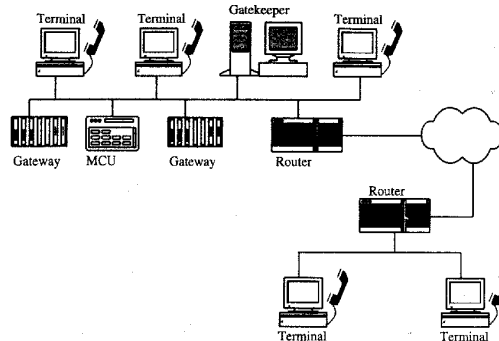
H.323 merupakan salah satu rekomendasi ITU yang mengkhususkan diri pada sebuah arsitektur dan metodologi keseluruhan dan yang bekerja bersama rekomendasi-rekomendasi lain.

Gambaran lingkup H.323 pada gambar 2.1. Arsitekturnya meliputi terminal H.323, *gateway*, *gatekeeper*, dan *Multipoint Controller Units (MCUs)*.



Gambar 2.1 Lingkup H.323

Gambar 2.2 menunjukkan sebuah representasi dari zona tersebut. Sebuah zona dapat menjangkau berbagai jaringan atau sub-jaringan, dan semuanya dalam sebuah zona yang tidak harus terus berdampingan.



Gambar 2.2 Contoh dari zona H.323

2.2 Gambaran Sinyal H.323

Gambar 2.4 menunjukkan setumpuk protokol H.323 pada pengujiannya. Dalam sebuah jaringan IP, istilah-istilah ini merujuk pada "*Transport Control Protokol (TCP)*" dan "*User Datagram Protokol (UDP)*". Terlihat jelas digambar, pertukaran media dilakukan dengan menggunakan RTP diluar UDP dan tentu saja, kapanpun RTP ada, RTCP-nya juga ada. Pada gambar 2.4, ditemukan dua protokol yang belum dibahas, yaitu : H.225.0 dan H.245. Dua protokol ini menentukan pesan-pesan aktual yang ditukar

antar *endpoint* H.323. Kedua protokol itu adalah protokol generik karena dapat digunakan diberbagai arsitektur jaringan. Ketika sampai pada arsitektur jaringan H.323, cara-cara dimana protokol-protokol H.225.0 dan H.245 dipakai, dikhususkan dengan rekomendasi H.323.

2.2.1 Gambaran Protokol-protokol H.323

Seperti yang telah disebutkan sebelumnya, pesan-pesan yang ditukar antara kesatuan H.323 dispesifikasi dalam rekomendasi ITU H.225.0 dan H.245.

H.245 merupakan sebuah protokol kontrol yang digunakan diantara dua *endpoint* atau lebih.

2.3 Alamat H. 323

Setiap kesatuan dalam jaringan H. 323 memiliki sebuah alamat jaringan yang secara khas mengidentifikasi kesatuan khusus tersebut. Dalam sebuah lingkungan IP, alamat jaringan adalah sebuah alamat IP.

2.4 Codec

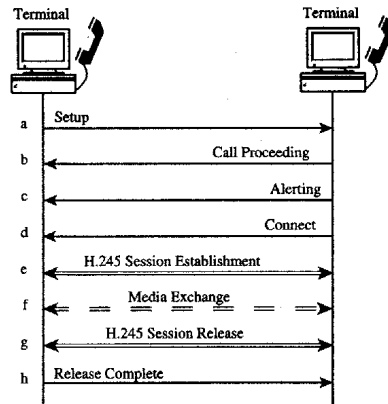
Pada gambar 2.4 codec audio dan video. Dukungan video merupakan pilihan. Dimana video didukung, maka sebuah *endpoint* H.323 (pada tingkat minimum) harus mendukung video yang berdasarkan pada H.261 *Quarter Common Intermediate Format (QCIF)*.

Audio / Video Application		Terminal / Application Control		
Audio / Video Codes	RTCP	H.225.0 RAS Signaling	H.225.0 Call Signaling	H.245 Control Signaling
RTP				
Unreliable Transport			Reliable Transport	
Network Layer (IP)				
Data Link Layer				
Physical Layer				

Gambar 2.4 Protokol stack pada H.323

2.5 Panggilan Dasar Tanpa Gatekeeper

Gatekeeper merupakan kesatuan pilihan dalam H.323. Dalam ke tidak hadir *gatekeeper*, sinyal panggil langsung terjadi diantara *endpoint*. Gambar 2.5 menunjukkan pemunculan dan penghapusan panggilan khusus dalam ketiadaan *gatekeeper*.



Gambar 2.5 Dasar panggilan tanpa menggunakan *gatekeeper*

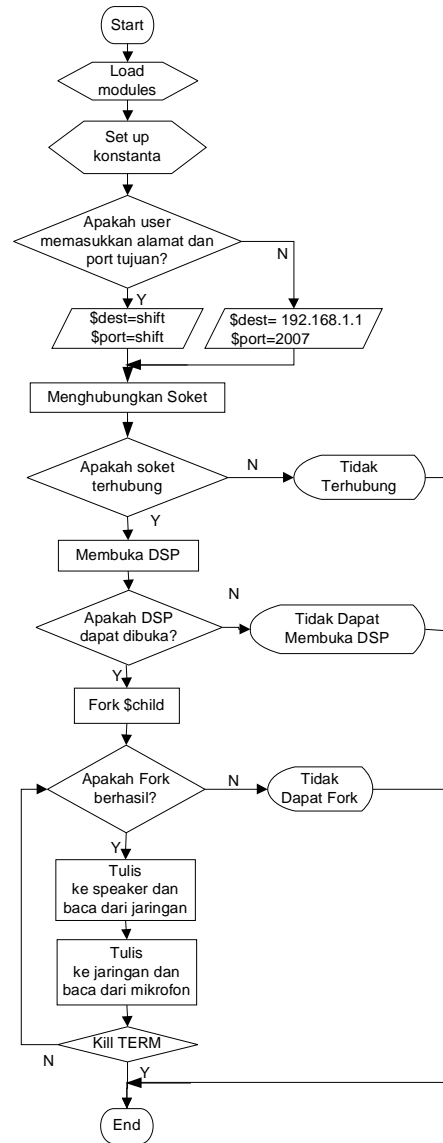
3. PERANCANGAN PERANGKAT LUNAK

Bab III membahas perancangan aplikasi voice chat yang terdiri dari perancangan perangkat lunak dengan diagram blok, hasil perancangan perangkat lunak, implementasi perangkat lunak.

3.1. Perancangan Perangkat Lunak dengan Diagram Blok

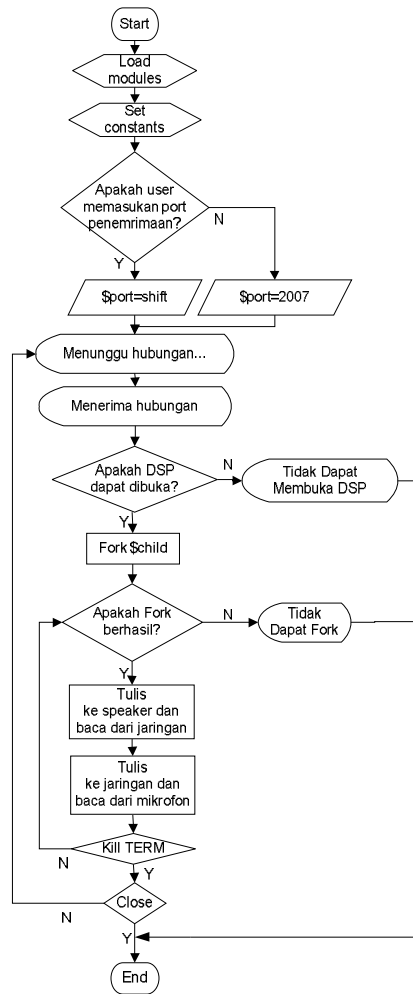
3.1.1 Diagram Blok Kirim.pl

Gambar 3.1 merupakan diagram blok untuk melakukan komunikasi dengan aplikasi *voice chat* yang dilakukan oleh pengirim.



Gambar 3.1 Diagram kirim.pl

3.1.2 Diagram Blok Terima.pl

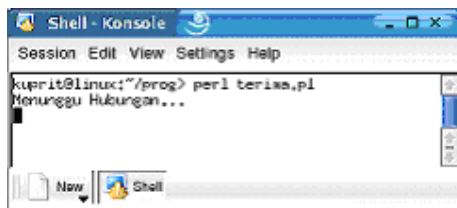


Gambar 3.2 Diagram terima.pl

Gambar 3.2 merupakan diagram blok untuk penerima.

4 Hasil Pengujian

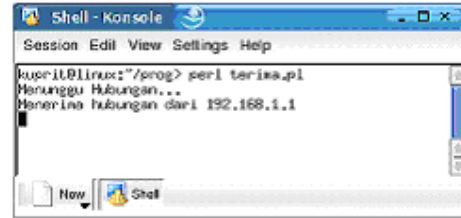
4.1 Pengujian Menggunakan Suara Dengan Dua Komputer



Gambar 4.3 Memperllihatkan aktivitas menunggu hubungan dari komputer kedua.

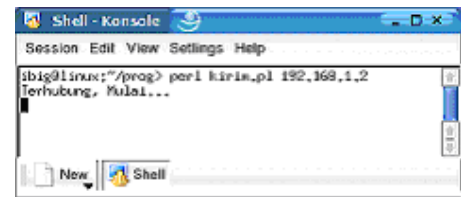
Pada pengujian yang kedua ini aplikasi voice chat diinstal di kedua buah komputer. Gambar 4.3 konsole pada komputer

kedua, aplikasi voice chat dijalankan untuk menunggu hubungan dari komputer pertama.



Gambar 4.4 Memperllihatkan aktivitas menerima hubungan dari komputer pertama.

Tampilan konsole Gambar 4.4 menerima hubungan dari komputer pertama dengan alamat IP 192.168.1.1



Gambar 4.5 Memperllihatkan aktivitas terjalin hubungan dengan komputer kedua.

Gambar 4.5 konsole pada komputer pertama, mengetik perintah perl kirim.pl diikuti dengan alamat IP komputer yang akan dituju. kemudian mulailah saling kirim pesan. Bila sudah terjalin hubungan dengan komputer kedua maka akan tertera pesan " Terhubung, mulai".

5 Penutup

5.1 Kesimpulan

Dari hasil pengujian terhadap program aplikasi maka dapat ditarik kesimpulan sebagai berikut :

1. *Delay* terkecil yang dicapai adalah 543 ms pada percobaan dengan *buffer* sebesar 1 KB.
2. Besarnya *delay* karena suara tidak dikompresi.
3. Semakin besar *buffer*, maka semakin besar *delay* yang ditimbulkan. Hal ini disebabkan besar *buffer* akan mempengaruhi beban pada jaringan, semakin besar *buffer* makan akan bertambah pula beban pada jaringan

5.2 Saran

Terdapat beberapa hal yang harus diperhatikan dalam menerapkan program aplikasi, yaitu:

1. Program ini masih terpisah, maka sebaiknya dijadikan satu program yang dapat melayani program tambahan untuk mengontrol program antar muka pengguna.
2. Program ini harus menghidupkan *receiver* sebelum menerima pengiriman dari client.

Dengan dibuat antar muka pengguna maka program ini dapat menerima pengirim suara dari *client* secara otomatis.

3. Memberi isyarat apabila terjadi pengiriman suara dari *client* atau panggilan telepon.
4. Dalam pengembangannya dapat digunakan untuk *teleconferencing*.
5. Toleransi pelan kerasnya suara diperhitungkan (jika toleransi suara terlalu rendah, maka suara tidak usah dikirim), maka membuat efisiensi pengiriman suara menjadi lebih baik.

Perlu mencari kecepatan yang sesuai pada jaringan terkoneksi yang akan dibuat.

Daftar Pustaka

1. Clinton Pierce, "Sams Teach Yourself Perl in 24 Hours", Sams, 2000.
2. Daniel Collins, "Carrier Grade Voice over IP", Mc Graw Hill, 2001.
3. Jonathan Davidson, "Voice over IP Fundamentals", Cisco Press, 2000.
4. Kadir Abdul, "Dasar Pemrograman Perl", Andi, 2002.
5. Kadir Abdul, " Pengenalan Unix dan Linux", Andi, 2002.
6. Tabratas Tharom, "Teknis dan Bisnis VoIP", PT Elex Media Komputindo, 2002.
7. <http://www.alsa-project.org>.
8. <http://www.cpan.org/>
9.<http://www.elektroindonesia.com/elektro/komp35a.html>.
10.<http://www.gematel.com/Edisi29/Artikel%20Lepas/lepas3.html>.
11.http://www.geocities.com/sistel_service/voip.htm.
12.<http://www.ilmukomputer.com>.
13.<http://www.komunikasi.org/voip/index.php3>.
14. <http://www.opensound.com>.
15. <http://www.perldoc.com/>
16.http://www.samag.com/tpj/issues/vo15_3.

Penulis

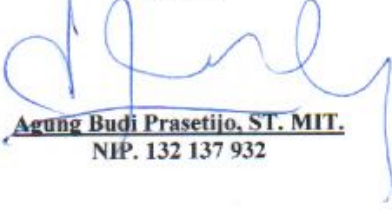


Fajar Adi Daniarto

L2F 097 634

Mahasiswa Teknik Elektro
Universitas Diponegoro,
Konsentrasi Informatika
Komputer, angkatan 1997.

Disetujui,



Agung Budi Prasetyo, ST. MIT.
NIP. 132 137 932